# Negotiation-based Price Discrimination for Information Goods

Gabi Koifman
Technion, Israel Institute of Technology
Haifa 32000, ISRAEL
kgabi@tx.technion.ac.il

Onn Shehory
IBM Haifa Research Lab, Haifa University
Haifa 31905, ISRAEL
onn@il.ibm.com

Avigdor Gal
Technion, Israel Institute of Technology
Haifa 32000, ISRAEL
avigal@ie.technion.ac.il

**Abstract**

*We present a mechanism for trading database tuples in a multiagent system. The mechanism enables negotiation and evaluation of database-based information goods. As part of the work we propose various policies for dynamic pricing of information goods. We have developed a test-bed that simulates a multi-agent system where agents use the offered mechanism and evaluated the system performance when sellers use different pricing policies in competitive and non-competitive environments. The investigated pricing policies include two novel pricing policies that implement negotiation and price discrimination across buyers. These were compared to two policies known in the art, which implement dynamic posted pricing. We have empirically demonstrated the superiority of the offered policies in maximizing sellers' gains. We have additionally identified equilibria profiles of these policies.*

## 1. Introduction

The rapid growth in electronic commerce has resulted in multiple buyers purchasing goods via the Internet, some of which use software agents for automated purchasing. Kephart et al. [7] envisioned that the global economy and the Internet would merge into an *information economy* bustling with billions of agents that exchange information goods. We envision that these agents will come to hold databases with valuable information gathered during their web activity. Such data may be valuable for other agents that would prefer purchasing it as an alternative to tedious data gathering from multiple sources. To do so, agents need to trade database tuples. This paper provides means for agents to do exactly that.

Trading database-tuples, much like the trade of any good, consists of several phases: the compatibility evaluation phase, where the buyer decides on the relevancy of the offered good; the utility evaluation phase, where the buyer estimates the gained utility from purchasing the good; the purchase phase, where some pricing mechanism is adopted, such as posted pricing or negotiation-based pricing; and, the post-purchase phase, where the transaction is completed and the good is transferred to the buyer.

The trade of database-based (DB-based hereinafter) information goods (i.e., DB tuples) introduces problems that are not present in the trade of traditional goods. Firstly, DBs are traditionally described by their schema. Therefore, the complexity of the compatibility evaluation phase, when purchasing DB tuples, stems from the need to generate a correct mapping between the buyer's DB schema and the seller's DB schema. Schema mapping between two DB schemas can be performed by considering schema information (e.g., attributes' description, attributes domain and constraints) and DB content. These are unique to the databases domain. Secondly, information goods (including DB-based goods) are experience goods [12]. That is, they can be valued only by knowing their content, which is revealed only once the good is transacted. To avoid losses, sellers have to advertise the goods in a way that minimizes the data revealed, nevertheless allowing buyers to understand the good's essence. In the DB-based information goods domain, the utility evaluation phase can be mapped to the problem of evaluating the number of tuples that the seller holds and the buyer does not.

Solutions to these problems, namely the schema compatibility checking and utility estimation problems, are presented in a separate document of ours [8]. Given these solutions, this paper addresses the complementary problems of pricing of and negotiation over (DB-based) information goods. Information goods are usually costly to produce, however reproduction is inexpensive, and the marginal cost is practically zero [12][13]. It is unlikely that prices, which, in traditional markets, are driven down to their marginal cost, will behave the same for information goods. Therefore, pricing information goods becomes a nontrivial task. Although we study the trade of DB-based information goods, the pricing problem we address is pricing of information goods, not necessarily in the context of DBs. We devise a negotiation model as well as two novel pricing policies. Our suggested policies, implementing price and product discrimination, are compared to policies known in the art. They experimentally prove superiority in maximizing sellers' gains and in providing market stability.

We begin by presenting the problem in Section 2. Section 3 describes supportive methodologies used. The negotiation model, including protocol and agents' reasoning model, is presented in Section 4. Section 5 details the experimental evaluation and the simulation results. In Section 6 we present related research, and Section 7 concludes.

## 2. Problem Description

The trade of DB-based information goods introduces pricing and negotiation problems. Given a set of economically motivated information agents residing on the Internet; given that each agent holds and maintains a DB; given that all DBs are in the same domain, DBs typically have different schemas, however some information overlap between them may exist. Additionally, given that agents need to acquire information they do not hold, and that information might reside in other agents' DBs. The problem we address is of providing the agents with means for: 1. negotiating the purchase of DB tuples; 2. buyers' bidding for the good; 3. sellers' pricing of the good. The bidding and pricing problems are typical to negotiation in general and agent negotiation in electronic markets in particular. However, the nature of the negotiated (DB-based) information goods is unique, and thus existing negotiation models, dealing with traditional products, do not suffice. Further, existing pricing mechanisms for information goods do not maximize (seller) profits.

## 3. Supportive Methodologies

This section presents the methodologies used in conjunction with the negotiation model proposed here. For clarity, we first briefly present the negotiation protocol. The protocol starts with sellers publishing themselves to buyers. A buyer may begin a negotiation process with the seller. The seller then sends an initial offer that includes relevant information on its goods (its schema and the number of tuples it holds). The buyer then starts a point-to-point negotiation process with the seller, during which the buyer first determines compatibility level and estimated utility from purchasing the good (Section 3.1). Once the buyer concludes that it wishes to purchase the seller's database tuples it begins negotiating over price and delivery time by exchanging offer and counter-offer messages. During the price negotiation process the seller uses a Bayesian learning mechanism to enable it to price the goods according to the buyer's assumed AL (Section 3.2.1), and learns the market trends (Section 3.2.2) to price the goods according to market trends when using one of the offered pricing policies.

### 3.1 Compatibility and Utility Evaluation

Because information goods are experience goods, buyers must adopt some compatibility checking mechanism to evaluate offered information goods and decide whether they are the goods they seek, without having the seller reveal the goods' content. In the specific case of DB-based information goods, the compatibility evaluation problem is reduced to the schema-mapping problem, that is, to providing a mapping between two schemas [4]. In a complementary study, we have devised a technique for generating schema mappings that exploits statistical information of the seller's database [8]. The technique assumes that the DBs are large enough such that, if two attributes match, their values are expected to hold a similar distribution. The statistical information required for that technique is revealed by the seller during the negotiation process, and used by the buyer to test whether the values of the seller's DB are sufficiently similar to the buyer's DB values. The estimation of the buyer's utility from purchasing DB tuples is performed by calculating a distance(seller, buyer) metric which indicates the number of distinct tuples that appear in the seller's DB but lack in the buyer's DB. For space reasons, the schema matching technique is not presented in this paper.

### 3.2 Learning

An agent's performance can be improved when it learns from prior encounters, where applicable. There are numerous methods for learning (e.g., [10][14]). Our model implements learning of buyers' anxiousness level (denoted as AL), and market trends. The former is used to differentiate among buyers, to perform price discrimination among them, and the latter is used for a pricing policy that considers market trends.

**3.2.1 Learning Buyer's AL.** As stated above, traditional schemes are impractical for pricing information goods. Varian [12] suggests that information goods be priced according to their value and not according to their cost. Information goods are highly differentiable, hence producers can use price discrimination and quality discrimination, as different groups of buyers are willing to pay different prices. The pricing mechanism suggested in this paper utilizes this differentiability.

Assuming a market where some buyers are more anxious than others, estimating the buyer's AL can be useful for a seller to perform discrimination and increase its profit. Price discrimination introduces two major difficulties, though. One is to determine the buyer's AL, and the other is to prevent buyers with high AL (i.e., high-end buyers) to purchase products at prices intended for buyers with low AL (i.e., low-end buyers) [13]. The question is, why would a buyer expose its true AL?

Buyers may try to manipulate sellers by pretending to be less anxious and thus, be offered lower prices. To be able to discriminate among buyers, a seller has to discriminate among products [3]. We suggest discrimination among information goods according to Quality of Service (QoS), and in particular the Time to Deliver (TTD) QoS. Likely, a very anxious buyer of information goods needs to have the information goods sooner than a less anxious buyer. We suggest inferring the buyer's AL via its TTD and price bid, thus price the good according to the inferred AL. Additionally, high-end buyers would not buy goods intended for low-end buyers, because the QoS of such good does not satisfy high-end buyers' TTD. Our model adopts the sequential decision making model offered in [14] in order to learn the buyer's AL. Learning is achieved by explicitly modeling beliefs about the negotiation environment and the participants under a probabilistic framework using Bayesian learning representation and update mechanism. For simplicity, we represent the AL by a discrete set of values, $AL \in \{AL\} = \{1,2,3,4,5\}$, AL=1 is the lowest AL, AL=5 the highest AL, the other values refer to intermediate ALs. Similarly, $TTD \in \{0,7,14,21,30\}$, values given in days. When a seller first encounters a buyer it assumes equal probabilities for each $AL \in \{AL\}$. That is, $p(Hi)=0.2\ \forall i \in \{AL\}$ where Hi is the hypothesis that the buyer has AL=i. Counter-offers sent from a buyer during a negotiation process are used by the seller to update its p(Hi). A counter-offer consists of a price bid and a TTD bid, e.g., <100$, 30 [days]>. The seller holds some assumptions, regarding the AL distribution of the buyers' population, translated into a probability table that is used in the Bayesian learning update mechanism. With Bayesian learning, the seller uses an incoming counter-offer (*e*) to update p(Hi): $P(Hi \setminus e) = \frac{P(e \setminus Hi) * P(Hi)}{\sum_j P(e \setminus Hj) * P(Hj)}$. Once the probabilities are updated, the seller draws lots according to the updated hypothesis distribution to decide what the buyer's AL is, and sends a response accordingly.

**3.2.2 Learning Market Trends.** A pricing policy that considers market trends may allow price optimization. To allow market-based pricing, a seller should maintain estimations of the current supply (denoted by *S*) and demand (denoted by *D*) levels for the good it sells and for interchangeable goods. For simplicity, we categorize *S*, *D* levels as low, normal and high. In our model, a seller estimates the *D* by the number of its sales. It estimates the *S* by counting the number of advertisements of interchangeable goods in the market. These estimates are used to evaluate the market trend. Initial reference levels of *S*, *D* are computed during a training session of the market. Then, time is partitioned into frames, and at the end of each frame, the seller evaluates *S*, *D* levels for that frame as follow: if *S*, *D* in the last frame is higher than the reference *S*, *D*, set level to high; if lower, set level to low; else, set level to normal. Using *S* and *D* levels, the seller updates its price-list. The price of a good increases as *S* decreases and *D* increases, and the price decreases as *S* increases and *D* decreases. After several consecutive frames (arbitrarily set to 4) the seller re-evaluates the reference *S* and *D*, to allow for a more up-to-date estimation of frame-to-frame supply and demand.

### 3.3 Pricing Policies

We evaluate four pricing policies. The first two, Trial-And-Error (TA) and Derivative Follower (DF), were suggested by Kephart et al. [7][11]. We believe that these pricing methods are naïve and that other pricing policies can perform better. The other two pricing policies we offer are Personalized Pricing (PP) and Market Based Personalized Pricing (MBPP). The policies differ significantly, as follows: the former two policies do not implement price discrimination, while the latter two do (based on TTD); the first two policies implement posted pricing, that is, a take-it-or-leave it scheme, and the other two enable negotiation between buyers and sellers. Detailed description of the pricing policies follows.

*TA* randomly generates prices according to the market price distribution g(p). *DF* experiments with incremental increases (or decreases) when the profitability level increases (decreases). Not implementing discrimination, each seller offers a single QoS(TTD), randomly selected from $\{0,7,14,21,30\}$. *PP* and *MBPP* are negotiation-based pricing policies. That is, the price can be changed during the negotiation process. In addition, both policies perform product differentiation by offering the good in several TTDs, which allows them to perform price discrimination. Each seller offers the good in all possible TTD values. Different prices are assigned to different AL values, to allow pricing the goods according to the buyer's willingness to pay, such that if ALi<ALj then price[ALi]<price[ALj]. Each list-price is also associated with a different TTD value, assuming there is a correlation between the buyer's AL and its required TTD, i.e., the more anxious the buyer is, the sooner it requires the goods. A seller that adopts *PP* uses the list-price that is associated with the buyer's assumed AL to price the goods, as described in Section 3.2.1. *MBPP* is based on PP, with the following addition. When adopting *MBPP*, the seller learns the market trends as described in Section 3.2.2 and updates its price list accordingly. Therefore, the price list dynamically changes, according to market conditions.

## 4. Negotiation Model

To enable trading DB-based information goods we devise a negotiation model: negotiation protocols, negotiation objects (the issues over which agreement has to be reached), and agents' reasoning model.

## 4.1 Negotiation Protocols

The model consists of multiple agents, denoted as players, situated in the DB-based information goods market, and an additional agent that serves as the mediator, denoted as the Database Exchange (DBE) agent. Each player holds two roles, seller and buyer, and has an initial DB. Sellers can benefit from selling DB tuples and buyers can benefit from expanding their DB. A buyer subscribes to a specific domain available at the DBE, thus receives the advertisements that sellers at that domain send to the DBE. Given a received seller's advertisement, a buyer may initiate negotiation with it. A buyer may manage concurrent negotiation processes to accelerate activity and allow surveying several alternatives before making the purchase decision. A seller may adopt various pricing policies.

The protocol is as follows. Sellers publish themselves to potential buyers using the DBE by sending a RequestToPublish message to it, the DBE sends a PublishingSeller message to the players that are subscribed to the seller's domain. A buyer may begin a negotiation process by sending, directly to the seller, a WillingToNegotiate message. The seller then sends an InitialOffer that includes relevant information on its goods (schema, number of tuples). The buyer then starts a point-to-point negotiation process with the seller, during which the buyer first determines compatibility level and estimated utility (Section 3.1). Once the buyer reaches the conclusion that it wishes to purchase the seller's DB tuples it begins negotiating over the price and TTD with the seller by exchanging CounterOffer messages. During the price negotiation process the seller uses a Bayesian learning mechanism to price the goods according to the buyer's assumed AL (Section 3.2.1), and learns the market trends (Section 3.2.2) to price the goods according to market trends when using *MBPP*. The negotiation ends either with agreement (the good then sent using TransferGoods message), or with either of the agents sending a TerminateNegotiation message. There are two issues over which agreement has to be reached (i.e., negotiation objects) in this negotiation model: the price of information goods, and their delivery time (TTD).

## 4.2 Agents Reasoning Model

**4.2.1 Buyer's bidding decision-making.** Denote a bid for a TTD by *TTDb*, a bid for price by *price*, a buyer's bid is a tuple of <*TTDb*, *price*>. *price* cannot exceed the buyer's budget for the transaction (denoted $\tau$). *TTDb* does not need to be the true TTD (denoted tTTD) of the buyer's, although eventually *TTDb* will converge to tTTD.[1] Note that the buyer's bidding decision-making is relevant only when agents adopt *PP* or *MBPP,* where negotiation is allowed. The buyer adopts the following assumptions: (1) A higher AL justifies a higher *price* and shorter tTTD and thus, shorter *TTDb*s; (2) It is unreasonable to bid *TTDb* shorter than tTTD, as the price for a shorter *TTDb* is likely higher; (3) A tTTD bid should have a probability greater than the probabilities of other *TTDb*s; (4) The probability to bid tTTD should increase as AL decreases; (5) It may be beneficial to pretend to have a reduced AL, to receive better terms. It thus follows that *TTDb* ≥ tTTD, and *price* $\leq \tau$. At each iteration of the negotiation process, the probability of *TTDb* = tTTD increases.

We distinguish between the first *TTDb* initiated by the buyer, and following *TTDb*s, which are a response to the seller's CounterOffer <*T,P*>. The buyer randomly generates the first *TTDb* according to a distribution that satisfies the model assumptions.

The buyer uses the following heuristic to determine its CounterOffer bid, distinguishing among four cases: (1) If $P \leq \tau$, $T$>tTTD, try to bid again for TTD, keep the price unchanged; (2) If $P \leq \tau$, $T$<tTTD, try to bid again for TTD and a lower price. The new price bid is lower than the previous bid by a randomly selected discount factor; (3) If $P > \tau$, $T$≤tTTD, try to bid again for price, by trying to split the difference, and maintaining the same TTD bid; (4) If $P > \tau$, $T$>tTTD, bid again for TTD and try to compromise on price by bidding for a lower price using a randomly selected discount factor;(5) Otherwise, $P \leq \tau$, $T$=tTTD, the buyer accepts the offer. Recall that in one-to-one negotiation there is no single, optimal point of price agreement. The whole range is valid, including the range of prices that are lower than the transaction budget, which is the possible domain of the buyer's price bids [5].

The buyer evaluates two conditions before placing a bid. Let $\theta$ denote the negotiation cost (i.e., communication cost), $\beta$ denote the average price paid for interchangeable goods. The first condition is that $\theta \leq 0.1^* \tau$. This limitation assures that the negotiation process is finite, as communication cost is finite. The 0.1 coefficient is chosen arbitrarily. Let h(AL) be a monotonic ascending function from AL to the reals, h(AL=0)=1. The second condition is that *price*$\leq \beta^*$h(AL). This assures that a buyer's willingness to pay higher prices is ascending with its anxiousness level. When one of these conditions evaluates to false, the buyer terminates the negotiation process. These conditions correspond to the buyer's goal to expand its DB at minimum cost, under the constraint imposed by its AL.

[1] We assume that, at the end of the negotiation, the tTTD should not be compromised. If *TTDb* is longer than tTTD, it is not sufficient, and the buyer will bid for a shorter TTD. If *TTDb* is shorter than tTTD – the buyer bids a lower *price* for tTTD.

**4.2.2 Seller's bidding decision making.** A seller's bid <*T*, *P*>, indicates that it is willing to sell its database tuples for *P*[$] and deliver it to the buyer in *T*[days]. The seller holds the following assumptions: (1) A negotiation process should be finite. Therefore, willingness to compromise the price should increase as time elapses. Thus, a price offer *price* should be accepted in encounter *t*, if $price \geq P*\gamma(t)$. $\gamma(t)$ is a factor that decreases over time, $\gamma(0)=1$. *t* indicates the number of encounters performed in a specific negotiation process; (2) Different TTD values may have different prices; (3) The TTD is an obligatory requirement of the buyer's.

When posted pricing is adopted, each seller offers a single QoS, denoted seller-TTD. The seller maintains a single price, denoted priceForAll, which is initially generated according to the market price distribution priceForAll~g(p). The CounterOffer <seller-TTD, priceForAll> is sent to the buyer one time per a negotiation process, as no negotiation is allowed. With DF, priceForAll is raised by δ% when a negotiation process ends with a purchase and is decreased by δ%, otherwise. With TA, the price is generated according to the market price distribution priceForAll~g(p).

When adopting a negotiation-based pricing policy, the seller prices the good at each iteration, that is, upon receiving a CounterOffer from the buyer. Let buyerOffer be the CounterOffer received from the buyer, consisting of a price bid and a TTD bid. Let list-price be an array of all available prices, where each price is associated with a TTD value. We denote an association between a list-price and a TTD by list-price[TTD]. If the CounterOffer satisfies the seller (i.e., buyerOffer.price ≥ list-price[buyerOffer.TTD]), then the seller accepts the offer. Otherwise, the seller estimates the buyer's AL using Learning Buyer's AL algorithm (Section 3.2.1) and sends CounterOffer<price[TTD (AL)], TTD (AL)>, where TTD (AL) is the associated TTD of AL. With *MBPP*, the seller follows the same rule as in PP, however the list-price value is dynamically changing following changes in market trends.

## 5. Experimental Evaluations

This work offers a model of the pricing of information goods. We test our policies via simulations. In the simulations, the sellers' utility (i.e., profit) is measured under various pricing policies and subject to different market settings. We seek a policy that exhibits the highest utility and is in equilibrium.

We have experimented with actual agents (i.e., autonomous SW entities encapsulating communication and reasoning models) that trade simulated goods, implementing the suggested negotiation model. The agents ran on three hosts connected via a LAN, thus exposed to distribution as well as communication constraints that are found in real electronic markets. Consequently, we expect the results to be rather close to those expected in actual markets. We have measured the profit of the sellers, when all sellers adopt the same pricing policy, under two market settings, namely competitive market and non-competitive market, and have identified policies that exhibit the highest profit. To identify equilibria, we performed experiments where a single agent deviated from the policy of the others. Comparison of the profit achieved by the deviating agent to the profit achieved by the others allowed us to identify policies which are in equilibrium. Below, we present the experimental settings, followed by the results.

### 5.1 Settings

Each experiment consisted of 30 players. We distinguish three types of settings: an experimental setting, an agent's setting and a system setting. The system's setting was fixed for all experiments, whereas an agent's setting was fixed within a set of experiments but not across sets. We have preformed several sets of experiments for each of the experimental setting, to allow statistical analysis.

An experimental setting consists of the following: (1) Assignment of a pricing policy to each agent. We have experimented with two types of assignments. The first is where all the agents adopt the same pricing policy (i.e., *homogenous)* and the second is where all agents, but one, adopt the same pricing policy (i.e., *heterogeneous*); (2) The type of market environment. **Non-competitive market** is a market where a buyer does not consider offers of other sellers before making a purchase decision, while **Competitive market** is a market where a buyer performs a price survey before making a purchasing decision. The buyer purchases the good from the seller that offers the good at the lowest tuple price among those surveyed. In our experiments, the buyer accumulates three acceptable counter-offers from three different sellers with whom it negotiates. It then continues the purchase process only with the seller that offers the lowest price, terminating the other two negotiation processes.

An agent's setting consists of the following: (1) *initial budget* ~Uniform [500, 2000]. These bounds are set arbitrarily to simulate real-life budget constrains; (2) price-list, an array of five prices, each price associated with an AL value, such that list-price[i] is associated with AL=i, such that if $AL_i<AL_j$ then $price[AL_i]<price[AL_j]$. This is accomplished using list-price[0]~uniform [0,20] (i.e., price for AL=0) and list-price[i+1]=list-price[i]*α where α~uniform[1,2]; (3) AL, randomly generated from the domain {1,2,3,4,5}; (4) *transaction budget*, which is the maximum budget that the buyer is willing to spend on a purchase. We chose a uniform distribution between 1 and 100 for this parameter. These bounds are set arbitrarily. Other bounds would be applicable too.

The following general simulation settings were chosen: (1) A two minutes *publish interval*. During a *publish* *interval* two randomly selected sellers advertise themselves. This interval length was chosen after experimenting with several values. Higher values have often exhibited system idle time, and lower values have resulted with an overloaded system. (2) The cost of sending a message was set to 0.5, which is 1% of the average good price.

## 5.2 System Profit

We define *system profit* as the sum of the transactions that were performed in the system. *System volume* is the cumulative number of transactions that were performed in the market. To compare performance of the pricing policies, we have measured the system profit and system volume during 6000 encounters in several experiments. We examined eight experimental settings. For each market environment type, we have used four different pricing policies assignments (one for each studied policy), all heterogeneous. The results (Figure 1) show that if all agents indeed adopt the same pricing policy, the *MBPP* policy exhibits the highest system profit in a non-competitive environment and the *PP* policy exhibits the highest system profit in a competitive environment. In both market environments *MBPP* and *PP* are superior to TA and DF. Although system profit is not an agent's profit but a cumulative profit, it is useful for estimating the average profit of a randomly selected agent when buyers' preferences are heterogeneous and sellers' prices are distributed uniformly.

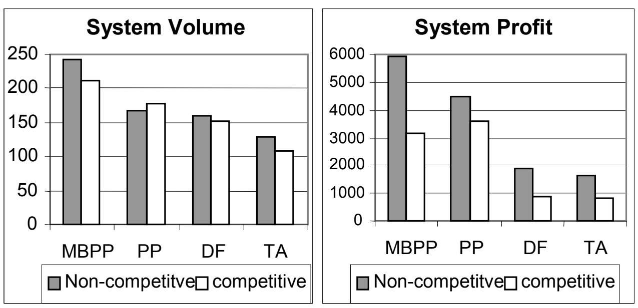


**Figure 1: System Profit and System Volume in competitive/non-competitive market**

**5.2.1 Competitive market vs. non-competitive market.** For all pricing policies, the profit in the non-competitive environment is higher than the profit in the competitive environment. The reason for this is that in a competitive market, according to the offered model, a buyer purchases the good at the lowest price among three offered prices. Let *p* be a random variable that represents the price, *f(p)* is the distribution function of *p*, and $p_1,p_2,p_3 \sim f(p)$. The purchase price in the competitive environment is $\min(p_1,p_2,p_3)$. It is trivial that the expected value of the purchase price in a competitive environment is lower than in a non-competitive environment. That is, $E(\min(p_1,p_2,p_3)) \leq E(p_1)$. This implies that the sellers' profit in a competitive environment is lower than in a non-competitive environment.

**5.2.2 TA and DF vs. PP and MBPP.** The results show that both *TA* and *DF* exhibit a much lower *System profit* level than *MBPP* and *PP* in both market environments. The significant difference is due to the characteristics of the pricing policies. *TA* and *DF* policies are both posted pricing that offer the information good at one QoS and do not enable negotiation, whereas the *MBPP* and *PP* policies implement product differention and price discrimination according to TTD, and are based on negotiation. They allow negotiating both on price and on TTD. Price discrimination increases profitability as it allows the seller to price according the buyer's willingness to pay. In addition, providing only one QoS to the buyers does not enable selling to the entire buyers' population, as only buyers with the exact, or lower, QoS requirement consider purchasing. For example, a buyer that requires TTD=0, will not purchase the good offered at TTD=30, as a 30 days delay in receiving the good is likely unacceptable for such a buyer. Therefore, a seller that offers the good at all available QoS, has a higher probability that buyers find the offered terms acceptable, and thus increase profits. In addition, not enabling negotiating over price, but offering a fixed price per encounter, decreases the probability for the encounter to culminate in a purchase, and thus, posted pricing based policies exhibit lower profitability compared to the case of negotiation based pricing policies.

**5.2.3 MBPP vs. PP.** The results show (Figure 1) that in a non-competitive market *MBPP* exhibits higher *system profit* than *PP*, while in a competitive market, *PP* exhibits higher *system profit* than *MBPP*. *MBPP*, which is a market sensitive pricing policy, drives prices down when lower demand is observed, while *PP* maintains fixed prices. In a competitive market, where some acceptable deals are not concluded because of others providing a lower price, the *MBPP* seller reacts by lowering prices. Hence, prices show a general decreasing trend, and *system profit* decreases.

**5.2.4 DF vs. TA.** Our results are in agreement with the work of Kephart et al. [7][11]. *DF* exhibits slightly higher *system profit* than *TA* in both market environments. Further, *DF* exhibits a higher increase, with respect to the increase in *system profit*, in *system volume* than *TA* does. Intuitively, *DF* allows more purchases than *TA* as it lowers its prices when a negotiation is not concluded with a purchase. Decreasing the price increases the probability that a buyer will make a purchase in the next encounter, as some buyers may have avoided purchasing the good because it was too expensive.

### 5.3 Equilibrium

It is unlikely that all agents adopt the same pricing policy, unless it is an equilibrium setting. Hence, it is reasonable to inquire whether our results apply in a realistic setting where each agent may choose its preferred pricing policy. More specifically, we look for the pricing policy that is in experimental equilibrium[2], if adopted. That is, no agent would deviate from the chosen pricing policy as it provides it with the highest expected utility (profit). To examine equilibria points, we have performed experiments in a competitive environment, where agent deviation from non-equilibrium strategy profiles is expected. We have first tested whether *PP* is an equilibrium pricing policy, as it had exhibited the best results when all agents have adopted it. We ran experiments where one agent had deviated from *PP* policy while the other agents use *PP*. Table 1 presents the results. The right column presents the average profit of each of the 29 agents that use *PP* policy, and the middle column presents the profit of the agent who deviated from *PP*. Each row refers to a pricing policy the deviating agent adopts. We chose the deviating agent to be the agent with an average list-price. This was done because deviation has a different effect on sellers with different price-lists, and we are interested in the average case. For example, the agent with the lowest set of prices will probably experience a minor change as a result of deviating from *PP* to *MBPP* than the agent with the highest set of prices, because its price-list is more attractive than the highest set of prices.

| Deviation policy | Deviative agent | 29 PP |
|---|---|---|
| 1 MBPP | 150.22 | 125.96 |
| 1 TA | 23.47 | 127.73 |
| 1 DF | 26.53 | 112.35 |

**Table 1. Average agent profit: one agent deviates and 29 follow PP. (based on 8 experiments)**

The results, which are presented in Table 1, show that assuming all other agents use the *PP* policy, the agent should deviate and adopt *MBPP*, as the average profit increases to 150. Therefore, adopting *PP* by all the agents does not exhibit the equilibrium property.

| Deviation policy | Deviative agent | 29 MBPP |
|---|---|---|
| 1 PP | 70.02 | 118.11 |
| 1 TA | 20.09 | 106.52 |
| 1 DF | 25.62 | 116.52 |

**Table 2. Average agent profit: one agent deviates and 29 follow MBPP. (based on 8 experiments)**

Following this conclusion, we have evaluated whether *MBPP* policy exhibits equilibrium. The results are presented in Table 2. The results show that deviating from *MBPP* to each of the other investigated pricing policies results in a lower profit for the deviating agent (profit decreases in 40%, 81%, 78% if deviating to *PP, TA, DF* respectively). The conclusion is that assuming all other agents adopt *MBPP*, an agent should not deviate from this policy. We conclude that experimental equilibrium [5] is reached when all the agents adopt *MBPP* policy.

We explain these results in the context of our model. When all the other agents adopt *PP*, an agent that adopts *MBPP* has an advantage over the others as it is the only agent that adjusts its prices in response to market trends. This behavior increases the probability that buyers would purchase from it in the long term. Consider a scenario where the market shows decreasing trend of purchasing due to high prices. The *MBPP* agent adaptively decreases its prices in response to a decrease in its *personal demand*, while other agents hold their prices fixed. Eventually, the *MBPP* agent's price becomes attractive to some buyers. Hence, a purchase will be made from the *MBPP* agent, increasing its profit, and not from the other agents.

## 6. Related Work

Information economy was identified by many researches as a unique niche in economics. Pricing of information goods is discussed in [11][12][13], among others. Electronic marketplaces and pricing of goods by software agents was the topic of many studies. A variety of pricing mechanisms can be used by software agents, including posted pricing [1][7][11], one-one-negotiation [14] and numerous types of auctions [2]. The work most relevant to ours was conducted by Kephart et al. They have studied the collective behavior of agents that dynamically price information goods and services [7][11], much like our work. However, there are some important differences between that work and ours. In [11], sellers compete to provide buyers with a commodity. Four different policies were examined, all dynamically posted pricing, including *TA* and *DF* that were also studied here as reference. Two additional policies in [11] (namely, MY and GT) were not examined here since their pre-requirement for perfect knowledge is not available in our model, or in real markets. Both [11] and our work investigate a market with vertically differentiated goods. Kephart et al. have also experimented with non-vertically differentiated good [7] and their results were similar in both cases. Their results show that in a competitive market with five sellers, when all sellers adopt the same pricing policy, *DF* exhibits the highest profit. When four sellers adopt the same policy and the fifth seller is free to choose an independent policy, it should choose *MY* which undercuts *DF* [7][11]. This means that equilibrium is reached when all sellers adopt MY, though, the detailed information required by *MY* is unlikely to be obtained. There are some additional differences between Kephart's work [11] and our work: (1) Kephart's model allows all of

[2] An experimental equilibrium, as defined in [5], refers to an equilibrium which is measured experimentally with respect to a given set of strategies.

the buyers to be informed, using a bulletin board, about the price and quality of the offered goods. In our work, the price is not revealed by a bulletin board, but rather as part of a secure point-to-point negotiation between a seller and a potential buyer. This difference enables performing price differention not only across vertically differentiated goods (i.e., different prices for goods of different QoS), but also across different negotiation processes. That is, a good of specific QoS can be sold at two different prices in two different negotiation processes. This is the idea that drives the *PP* and *MBPP* policies that were proposed in this paper; (2) In [11], advertising the prices using a bulletin board enables the buyers to perform a full sellers prices-survey and make an educated decision as for the good with the lowest price. The assumption that a buyer has access to perfect, completely up-to-date information about the sellers' prices and qualities is crucial in [11]. This assumption does not hold in our work. One of the implications of this difference is, since prices are not publicly available in our work, buyers can discover the price offered by each seller only by engaging in a negotiation process with it. The cost involved in a negotiation process (e.g., communication cost and evaluation cost) prohibits the performance of entire market prices-survey. Therefore, a buyer surveys a limited number of sellers; (3) The proposed model extends the buyer's reasoning model suggested in [11]. In [11], a buyer purchases if the offered price and QoS are sufficient, while in our model we add additional conditions, including fairness (price should not be higher than what the buyer had already paid for other purchases), and value-added service constraint (i.e. *distance* measure). Jennings et al. proposed to characterize a negotiation model by its protocols, objects, and reasoning model [6]. We used this framework to describe the offered negotiation model. Kraus [9] described the major issues that should be addressed when designing an agent with negotiation capabilities, including, negotiation protocol, reasoning model, decision making procedures, and strategies to be employed by an agent. We addressed these issues when designing the offered negotiation model. Zeng and Sycara used learning to improve agents' negotiation competence [14]. They used the Bayasian framework to update the knowledge and belief that each agent has about the environment and other agents and empirically showed that learning is beneficial. We embedded the Bayesian learning ideas in the seller's protocol to enable it to learn true ALs of buyers.

## 7. Conclusions

In this study we have investigated the problem of agents trading DB-based information goods and have offered a negotiation mechanism that enables such trading. One of the goals of this research is to explore novel pricing policies for the information goods domain in an attempt to optimize sellers' profits. We have investigated two novel pricing policies and compared them to two policies that were offered in previous research. Via experiments performed in a realistic setting, we show that the new policies are superior to other pricing policies known in the art. In particular, in a competitive market the Personalized Pricing exhibits the highest system profit, while in a non-competitive market, Market Based Personalized Pricing exhibits the highest system profit. However, equilibrium in competitive market is achieved when *MBPP* is adopted by all agents. Therefore we conclude that, at least in electronic markets where information goods are traded, and in particular when prices are not common knowledge, pricing policies that allow negotiation and discrimination perform better than posted pricing policies. Additionally, sellers benefit from learning the environment and adapting their pricing policy accordingly.

## 8. References


[1] A. Chavez, P. Maes, Kasbah: An Agent Marketplace for Buying and Selling Goods. Proc. of PAAM'96, pp. 75-90, London, UK, April 1996.

[2] J. Collins, S. Jamison, B. Mobasher, M. Gini, A Market Architecture for Multi-Agent Contracting, Proc. of Agents'97, 285-292, St. Paul, Minnepolis,1997.

[3] A. Dedeke, Self-Selection Strategies for Information Goods. First Monday, volume 7, number 3,2002.

[4] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi - A Framework for Modeling and Evaluating Automatic Semantic Reconciliation. To appear in VLDB Journal, 2004.

[5] C. Goldman, S. Kraus, O. Shehory, On Experimental Equilibria Strategies for Selecting Sellers and Satisfying Buyers, Decision Support Systems Journal. 2003.

[6] N. Jennings, S. Parsons, C. Sierra, P. Faratin, Automated Negotiation, Proc. of PAAM'00, pp 23-30, UK ,2000.

[7] J. Kephart, J. Hanson, A. Greenwald, Dynamic pricing by software agents. Computer Networks 32(6),pp 731-752 ,2000.

[8] G. Koifman Multiagent negotiation over DB-based information goods, Msc. Thesis, Technion, Israel, 2004. available at kgabi@tx.techion.ac.il.

[9] S. Kraus, Automated Negotiation and Decision Making in Multiagent Environments. EASSS 2001, pp 150-172, Prague, Czech Republic, 2001.

[10] T.Mitchell, Machine Learning, Boston, McGraw-Hill, 1997.

[11] J. Sairamesh, J. O. Kephart, Price Dynamics of Vertically Differentiated Information Markets, Proc. of ICE'98, pp 28-36, Charleston, SC, USA, 1998.

[12] H. Varian. Markets for Information Goods. In proc.: Monetary Policy in a World of Knowledge-Based Growth, Quality Change, and Uncertain Measurement, 2000.

[13] H. Varian, Pricing information goods, Research Libraries Group Symposium on "Scholarship in the New Information Environment", Harvard Law School, May 2-3, 1995.

[14] D. Zeng, K. Sycara, Bayesian Learning in negotiation, Int. Journal of Human-Computer Studies (1998) 48, pp. 125-141.